Global Information Assurance Certification Paper

Practical Assignment SANS GIAC
D Mac leod
22-November-2000

Network Detects

Network Detect #1

Network Trace Of Occurrence

2000/03/10 20:43:23 OUT IN	**4000 UDP_Packet**	**source.sub.net.26**	**protected.sub.net.147 00123 00123 ntp**
2000/04/07 20:58:15 OUT IN	**4000 UDP_Packet**	**source.sub.net.26**	**protected.sub.net.186 00123 00123 ntp**
2000/04/03 19:43:16 OUT IN	**4000 UDP_Packet**	**source.sub.net.26**	**protected.sub.net.228 00123 00123 ntp**
2000/03/03 20:13:26 OUT IN	**4000 UDP_Packet**	**source.sub.net.27**	**protected.sub.net.149 00123 00123 ntp**
2000/03/13 21:52:09 OUT IN	**4000 UDP_Packet**	**source.sub.net.27**	**protected.sub.net.203 00123 00123 ntp**
2000/03/02 18:43:27 OUT IN	**4000 UDP_Packet**	**source.sub.net.27**	**protected.sub.net.234 00123 00123 ntp**
2000/04/04 19:19:53 OUT IN	**4000 UDP_Packet**	**source.sub.net.32**	**protected.sub.net.134 00123 00123 ntp**
2000/03/07 18:43:23 OUT IN	**4000 UDP_Packet**	**source.sub.net.32**	**protected.sub.net.197 00123 00123 ntp**
2000/03/13 20:58:21 OUT IN	**4000 UDP_Packet**	**source.sub.net.32**	**protected.sub.net.203 00123 00123 ntp**
2000/03/14 21:43:21 OUT IN	**4000 UDP_Packet**	**source.sub.net.37**	**protected.sub.net.155 00123 00123 ntp**
2000/03/06 19:24:45 OUT IN	**4000 UDP_Packet**	**source.sub.net.37**	**protected.sub.net.204 00123 00123 ntp**
2000/03/05 18:43:26 OUT IN	**4000 UDP_Packet**	**source.sub.net.37**	**protected.sub.net.226 00123 00123 ntp**
2000/04/03 20:13:13 OUT IN	**4000 UDP_Packet**	**source.sub.net.37**	**protected.sub.net.35 00123 00123 ntp**
2000/03/09 18:43:24 OUT IN	**4000 UDP_Packet**	**source.sub.net.38**	**protected.sub.net.116 00123 00123 ntp**
2000/03/04 22:58:25 OUT IN	**4000 UDP_Packet**	**source.sub.net.38**	**protected.sub.net.154 00123 00123 ntp**
2000/04/08 15:13:22 OUT IN	**4000 UDP_Packet**	**source.sub.net.38**	**protected.sub.net.217 00123 00123 ntp**
2000/03/08 19:36:57 OUT IN	**4000 UDP_Packet**	**source.sub.net.39**	**protected.sub.net.165 00123 00123 ntp**
2000/03/11 15:58:23 OUT IN	**4000 UDP_Packet**	**source.sub.net.39**	**protected.sub.net.181 00123 00123 ntp**
2000/04/06 19:21:21 OUT IN	**4000 UDP_Packet**	**source.sub.net.39**	**protected.sub.net.185 00123 00123 ntp**
2000/04/06 19:28:15 OUT IN	**4000 UDP_Packet**	**source.sub.net.39**	**protected.sub.net.185 00123 00123 ntp**

Source of occurrence

Sensor on a protected customer network

Type of network device that detected the occurrence

CISCO SecureIDS (Netranger)
Log file is in field-reduced format.

Indications, if any, that the IP source was spoofed

Not likely. The scan for the port would be executed in order to get information about the protected network. Spoofing the source makes it technically challenging to receive the results of the scan.

Attack Description

Scan for NTP, Network Time Protocol

Attack Mechanism

The trace is a scan for UDP port 123. Port 123 is normally associated with NTP, which is the network time protocol. NTP is a service for a client to synchronize their clock against a trusted source. This scan took place over a period of one month, a slow scan. The source port of the scan is always the same as the destination port. The source address of the scan is not a business partner of the protected network, nor as the protected network ever offered NTP service from any of the scanned hosts. The sporadic nature of the scan could indicate that one node that is only up periodically on the scanning network has a miss-configured NTP address, i.e. perhaps the originating node is programmed to look in one /24 network for its NTP services. There are no indications of any of the hosts in the protected network accessing any nodes on the scanning network before the scan took place. There are also no indications that any of the scanned hosts responded to the scan.

Correlations

Address Correlation

The customer in question operates several sensors on different networks. There were no indications of the source IP address in any of the other sensors A search for the source IP noting other detecting scans using www.google.com and the GIAC for them was also negative.

Attack Type Correlation

There are no indications for scanning of NTP in any of the other sensors operated by the customer. A search of the SANS GIAC and of the CERT/CC current activity show no reports of NTP scanning.

Evidence of active targeting of the monitored network

Possible. The scan was not detected on any other segments, and it was performed very slowly. However the customer does not now, not has ever provided NTP service, this indicates that source is not really scanning for NTP, or is not familiar with the customer setup.

Severity of the occurrence
Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(2 + 0) – (5 + 5) = -8

(The targets are not core servers + The attack seemed only to be recon in nature) – (NTP is not running + (The firewall blocked the scan + The IDS detected))

What is not properly indicated about this attack in the above formula is the very slow, determined nature of the scan. Adding this to the end of the formula on a scale from 0-5. 0 being light the IDS radar up like a Christmas tree, and 5 being a very slow, quiet scan, the new value is –3

((2 + 0) – (5 + 5)) + (5) = -3

Recommendations to defend against the occurrence

Current countermeasures were sufficient. A search of the xforce.iss.net and the www.securityfocus.com vulnerability databases show no known vulnerabilities in NTP, however Xforce entry 1814 notes that system information could be revealed back to the requestor if you are running the service.
The source could be shunned at the border router to prevent further network access. Communications with the customers CIRT would be a wise defense option to look for other seeing the same probe. The slow, deliberate nature of the scan indicates a sense of determination of the attacker. While the common Trojan lists do not indicate a port of 123 as being common, the protected network should be tested for the presence of a Trojan listening on the NTP port. Having the protected networks CIRT contact the source in question would be beneficial.

Related test question

Using a spoofed source packet in a reconnaissance scan is useful when

A You are attempting to inspect a packet in an already established data stream
B The spoofed packet goes along with your real address as a decoy
C The spoofed packet has artificially low TTL so the data will return to the real address
D All of the above

B

Network Detect #2

Network Trace Of Occurrence

Alert File

2000-11-17 18:11:18	**2002003**	**SNMP backdoor**	**24.source.X.X**	**MY.NODE**	**community=private**
2000-11-17 18:11:15	**2003401**	**SNMP port probe**	**24.source.X.X**	**MY.NODE**	

Packet Capture from Network Ice and viewed in ethereal

User Datagram Protocol
Source port: 2437 (2437)
Destination port: snmp (161)
Length: 50
Checksum: 0x6e33
Simple Network Management Protocol
Version: 1
Community: public
PDU type: GET
Request Id: 0x2539c
Error Status: NO ERROR
Error Index: 0
Object identifier 1: 1.3.6.1.2.1.1.2.0 (.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0)
Value: NULL

User Datagram Protocol
Source port: 2437 (2437)
Destination port: snmp (161)
Length: 51
Checksum: 0x0c2a
Simple Network Management Protocol
Version: 1
Community: private
PDU type: GET
Request Id: 0x25585
Error Status: NO ERROR

Error Index: 0
Object identifier 1: 1.3.6.1.2.1.1.2.0 (.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0)
Value: NULL

Source of occurrence

Sensor on a protected cable modem.

Type of network device that detected the occurrence

Network ICE Personal Firewall

Indications that the IP source was spoofed

Not likely. The scan for the port would be executed in order to get information about the protected node. Spoofing the source makes it technically challenging to receive the results of the scan.

Attack Description

A scan for SNMP and a scan for SNMP private community strings.

Attack Mechanism

The scan is first a probe for the SNMP service. The second is an attempt as SNMP community string guessing. SNMP, Simple Network Management Protocol, is a mechanism for controlling and getting information from network devices. Community strings are the SNMP version of authentication. The default community strings are PUBLIC for read and PRIVATE for write. Read access gives you the ability to get information from the network node, such as uptime, network traffic stats etc. Write access gives you the ability affect changes on the device, such as disabling a port. Many network management consoles, such as HPOpenview, use SNMP. It is possible that the node is a network monitoring station for an ISP that was miss-configured to scan the entire subnet, or that the ISP is taking an active stance on security or AUP enforcement is scanning their range. SNMP managed nodes are very common on a large cable modem LAN. It is possible that a malicious actor was attempting to connect to the infrastructure components that live on the 24 IP range not on the other more out of band private RFC 1918 addresses that cable modem LANS commonly use for infrastructure. The default SNMP password is number 10 on the SANS Top Internet Threat consensus

CVE Entries:
Guessable SNMP community name - CAN-1999-0516
Hidden SNMP community strings - CAN-1999-0254, CAN-1999-0186
Default or blank SNMP community name (public) - CAN-1999-0517

Additional Information can be found at
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid210315

There are other possibilities on what was being scanned for, such as www.securityfocus.com BUGTRAQ ID 177 and 1177, which describe SNMP software with vulnerabilities beyond the more common default password problems.

Correlations

Address Correlation

The node in question only has access to logs from one sensor, the personal firewall. A search for the source IP noting other detecting scans using www.google.com and the GIAC for them was also negative. Although the scan was very recent and it is possible that in the next few days others will report the same scanner IP.

Attack Type Correlation

SNMP scanning is very common. The more aggressive probes involving guessing of the community strings is less common, however still very prevalent. Many posts to GIAC demonstrate this.

http://www.sans.org/y2k/122799-10.htm

Dec 26 13:23:56 genesis portsentry[16354]: attackalert: Connect from host: 172.20.20.1/172.20.20.1 to UDP port: 162
Dec 26 13:23:56 genesis portsentry[16354]: attackalert: Ignoring UDP response per configuration file setting.

http://www.sans.org/y2k/practical/EAVazquezJr.html#detect2data

[**] SNMP public access [**]
05/29-16:58:21.047981 216.164.136.103:1029 -> 192.168.1.67:161
UDP TTL:49 TOS:0x0 ID:11015
Len: 51
[**] SNMP public access [**]
05/29-16:58:23.034753 216.164.136.103:1029 -> 192.168.1.67:161
UDP TTL:49 TOS:0x0 ID:11016
Len: 51

Evidence of active targeting of the monitored network

Likely not. This is cable modem LANd. Scans, probes, attacks and the such are part of living in this neighborhood. The protected node does not, nor has it ever, run SNMP. It is very likely if data was available for other nodes in the surrounding network the scan would be seen on all the other nodes.

Severity of the occurrence
Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(2 + 4) – (5 + 5) = -4

(The targets is simply a home PC + The attack could have given write access to network devices) – (SNMP is not running + (The firewall blocked the scan + The IDS detected))

Recommendations to defend against the occurrence

Current countermeasures were sufficient. Contacting the ISP about the scan would be a wise action for the node owner. The personal firewall has the ability to block intruders and this should be done. The PC could be moved behind a more robust firewall for greater security, but in this case, the systems upkeep and capital cost would not be justified. If an inexpensive ipchains firewall were to be implemented, the rule set for SNMP blocking would be:

ipchains –A input –i eth0 –p tcp -s 0.0.0.0/0 –d 0.0.0.0/0 161:162 -l –j DENY
ipchains –A input –i eth0 –p udp -s 0.0.0.0/0 –d 0.0.0.0/0 161:162 -l –j DENY

Related Test Question

When using SNMP on your internal network, you

A Must allow port 25 TCP and UDP through the firewall for proper connectivity.
B Should stop SNMP at your perimeter if at all possible.
C Change the default community strings.
D Use POP3 as a safer alterative to SNMP.
E B and C
F A and D

E

Network Detect #3

1999/11/02 9:32:51 PM GMT -0500: D-Link DE-530CT-D..[0000][No matching rule] Blocking incoming UDP: src=266.266.266.2, dst=protected.node, sport=4433, dport=4433.

1999/11/02 9:33:51 PM GMT -0500: D-Link DE-530CT-D..[0000][No matching rule] Blocking incoming UDP: src=266.266.266.2, dst=protected.node, sport=4433, dport=4433.

1999/11/02 9:34:20 PM GMT -0500: D-Link DE-530CT-D..[0000][No matching rule] Blocking incoming UDP: src=266.266.266.2, dst=protected.node, sport=4433, dport=4433

1999/11/02 9:35:44 PM GMT -0500: D-Link DE-530CT-D..[0000][No matching rule] Blocking incoming UDP: src=266.266.266.2, dst=protected.node, sport=4433, dport=4433.

…. Scan Continues for days

Source of occurrence

Sensor on a protected customer network

Type of network device that detected the occurrence

CONceal Personal Firewall.

Indications, if any, that the IP source was spoofed

None. The source was traced down in the end.

Attack Description

Scan for UDP 4433

Attack Mechanism

The trace is a UDP scan from one IP address to the protected node. It is always from source port 4433 and to destination port 4433. The frequency of the scans is every minute or so and the detect goes on for days. When the DNS lookup was completed against the source IP, it turned out to be the employers LAN. The customer runs software to allow remote connections to portions of the employers LAN. This VPN software runs on UDP 4433. The employee tunnel had

terminated non-cleanly and the VPN device was configured not to drop tunnels unless explicitly instructed. This trace is the result of the VPN device probing for the client end of the tunnel.

Correlations

Address Correlation

The node in question only has access to logs from one sensor, the personal firewall. A search for the source IP noting other detecting scans using www.google.com and the GIAC for them was also negative. When a DNS resolution was made of the IP, the IP was the business IP of the protected networks employer.

Attack Type Correlation

None. For completeness, there are no indications for scanning of 4433 in any of the other sensors operated by the customer. A search of the SANS GIAC and of the CERT/CC current activity show no reports of 4433 scanning.

Evidence of active targeting of the monitored network

As do most IDS detects, this detect turned out to be a false alarm.

Severity of the occurrence

Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(2 + 0) – (5 + 5) = -8

(The targets are not core servers + The attack seemed only to be recon in nature) – (4433 is not listening + The firewall blocked the scan)

The formula is completed for reasons of completeness, it was false positive, but it technically was still detected and stopped.

Recommendations to defend against the occurrence

Reconfigure the VPN device to timeout idle or dropped tunnels.

Current countermeasures were sufficient. The PC could be moved behind a more robust firewall for greater security, but in this case, the systems upkeep and capital cost would not be justified. If an inexpensive ipchains firewall were to be implemented, the rule set for SNMP blocking would be:

ipchains –A input –i eth0 –p udp -s 0.0.0.0/0 –d 0.0.0.0/0 4433:4433 -l –j DENY

However, as stated, this was a false positive, so this rule would disable the VPN software.

Related test question

One of the challenges of using and IDS in a network that uses VPN technology is

A Since the tunnel is encrypted, payload analysis cannot be done by the IDS outside the VPN gateway.
B Most tunnels use a secure form of key exchange for tunnel setup that conflicts with many IDS systems communications.
C Any IDS based in libpcap will need the most current version of the software installed.
D Once the IDS decrypts the tunnel, the session will be lost to the VPN client.

A

Network Detect #4

Network Trace Of Occurrence

eventdate	eventname	sourceport	destinationport	sourceaddress	destinationaddress
7/29/99 7:27:17	http_php_read	1718	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:18	http_sgi_wrap	1720	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:18	http_sg_wrap	1721	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:18	http_glimpse	1722	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:18	http_unix_passwords	1722	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:20	http_testcgi	1728	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:40:07	http_apache_dos	3183	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:14	http_phf	1652	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:17	http_unix_passwords	1718	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:17	http_nphtestcgi	1716	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:15	http_nphtestcgi	1676	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:15	http_unix_passwords	1672	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:15	http_phf	1672	http	sourcenet.sub.net	protected.sub.net
7/29/99 7:27:14	http_unix_password	1652	http	sourcenet.sub.net	protected.sub.net

7/29/99 7:27:13	**http_testcgi**	**1624**	**http**	**sourcenet.sub.net**	**protected.sub.net**
7/29/99 7:40:07	**http_apache_dos**	**3182**	**http**	**sourcenet.sub.net**	**protected.sub.net**

Source of occurrence

Sensor on a protected customer network

Type of network device that detected the occurrence

ISS Realsecure in field reduced format

Indications, if any, that the IP source was spoofed

Not likely. The scan for the port would be executed in order to get information about the protected network. Spoofing the source makes it technically challenging to receive the results of the scan.

Attack Description

A scan for vulnerable CGI applications.

Attack Mechanism

The scan works by requesting common known CGI files from a web server. An interesting note in an otherwise common scan is that two of the scans take place 13 minutes later. This would normally suggest the attacker received information back from the first scan that was then use to better target the second attack. However, since the machine scanned was not a web server, no information would have been returned to lead the attacker in that direction. Since that last two attacks are a denial of service attempt, as opposed to the access or information gathering of the initial scan, perhaps the attacker got angry that the server returned no vulnerabilities and decided to DoS it.

Correlations

Address Correlation

The customer in question operates several sensors on different networks. There were no indications of the source IP address in any of the other sensors A search for the source IP noting other detecting scans using www.google.com and the GIAC for them was also negative. However since the scan did not target just the web server, and it did not scan the entire subnet, it suggests confusion in targeting on the part of the adversary.

Attack Type Correlation

CGI scanning very common on the Internet.

http://www.sans.org/y2k/060600-1200.htm

206.105.207.203 - - [04/Jun/2000:23:48:38 -0400]
"GET /cgi-bin/phf HTTP/1.0" 404 279
206.105.207.203 - - [04/Jun/2000:23:49:15 -0400]
"GET /cgi-bin/htmlscript?../../../../etc/passwdHTTP/1.0" 404 -
[Sun Jun 4 23:48:38 2000] [error] [client 206.105.207.203]
script not found or unable to stat: /xxxxx/web/cgi-bin/phf
[Sun Jun 4 23:49:15 2000] [error] [client 206.105.207.203]
script not found or unable to stat: /xxxxx/web/cgi-bin/htmlscript

Many tools exist for scanning for exercising vulnerabilities on web servers. Some of the more common ones can be found at http://packetstorm.securify.com/UNIX/cgi-scanners/

Evidence of active targeting of the monitored network

Not likely. The server that was targeted with the CGI attacks is not, nor has it ever offered web services. What does add to the mystery is the fact that no other machine in the subnet was targeting, one might normally expect in such an attempt that does not specifically target the web server.

Severity of the occurrence
Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(5+ 0) – (5 + 3) = -3

(The target is a core server + The attack was for web, this is not a web) – (web is not running + The IDS detected the scan)

Recommendations to defend against the occurrence

Current countermeasures were sufficient. The regular scanning of your own web server and education of your development staff on the need for writing secure CGI apps is essential. This IP should be monitored for future activity. The archive logs should be reviewed also to see if any connections from this address were

made in the past to the systems. A note to the ISP with the logs and notifying the upstream cert would be advisable.

Inserting extra characters in the CGI stream to attempt to evade detection by the IDS is

A Impractical due to current TCP header length requirements for the HTTP/HEAD fetch command
B Impractical due to CGI scanners being a client run utility
C Can be effective since the web server will ignore certain extra characters
D Can be effective since the IDS will not assemble the corresponding ICMP-NOHEADER error code returned by the web server

Network Detect #5

Network Trace Of Occurrence

```
18:23:04.116311 P 64.88.246.111 > my.lan.255: icmp: echo request
18:23:04.116432 P 215.99.23.123 > my.lan.255: icmp: echo request
18:23:04.116537 P 112.76.112.8 > my.lan.255: icmp: echo request
18:23:04.116741 P 23.13.60.5 > my.lan.255: icmp: echo request
18:23:04.116847 P 175.67.14.12 > my.lan.255: icmp: echo request
18:23:04.116852 P 234.175.9.234 > my.lan.255: icmp: echo request
18:23:04.116935 P 76.65.185.154 > my.lan.255: icmp: echo request
18:23:05.127063 P 240.179.143.67 > my.lan.255: icmp: echo request
18:23:05.127162 P 66.163.53.54 > my.lan.255: icmp: echo request
18:23:05.127243 P 236.31.45.98 > my.lan.255: icmp: echo request

…. This continues for ~4 hours with the same pattern of random IP sources.
```

Source of occurrence

Sensor on a protected customer network

Type of network device that detected the occurrence

SHADOW tcpdump sensor box.

Indications, if any, that the IP source was spoofed

Very Likely. This appeared to be a saturation style denial of service attack. The different addresses which appear as the source indicate that the attacker is possibly generating pseduo-random IP addresses. The denial of service saturation attacks will work without any return information being required by the attacking machine, thereby negating the requirement to use their real IP address.

Attack Description

ICMP Flood Attack

Attack Mechanism

The trace is an example of an ICMP echo request flood. The originating machine uses a tool to create spoofed ping packets and streams them out to the victim LAN. By sending the ping packets to the /24 broadcast, they are hoping to cause all the machines on the victim LAN to respond with an icmp echo reply. This LAN stops directed broadcasts at a down stream filtering device so there was no indication of any of the protected machines responding to the attack. In this case the originating network did not have the capacity to have a noticeable affect on the protected network so the attack was not successful. The fact that the LAN did not get saturated by the attack, leaves the protected company to believe it was not a DDOS army, but rather a single host that was flooding the net.

Correlations

Address Correlation

The customer in question operates several sensors on different networks. There were no indications of the source IP address in any of the other sensors A search for the source IP noting other detecting scans using www.google.com and the GIAC for them was also negative. Searching for the source IP address would be of little use given the very high likely hood that they are spoofed.

Attack Type Correlation

Saturation style denials of service attacks have amazing attention on the Internet currently. Their distributed cousins are a subject of much research. A google search on denial of service will result in much reading for the searcher.

Evidence of active targeting of the monitored network

Possible. The ICMP echo replies were sent to the customers address in flood style in all likely hood to disable the network. In an attack designed to gain access,

there is some benefit to the attacker to randomly scanning the Internet looking for hosts, as they might find them and use them as launching points for other targeted activity. Randomly picking IP ranges to saturation flood, while a pass time no doubt for the severely bored, seem less likely.

Severity of the occurrence
Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(3 + 3) – (0 + 3) = 0

(Some targets are core servers + The data volume was not sufficient) - (None + The LAN did not respond to .255)

Recommendations to defend against the occurrence

The Consensus Roadmap for Defeating Distributed Denial of Service Attacks http://www.sans.org/ddos_roadmap.htm contains excellent advice on what individual sights can to prevent DDOS attacks from occurring from their network It also contains advice for what to do if you are the victim of such an attack. DDOS attacks are the subject of much research in the community on ways to help defeat them. Saturation attacks are difficult to defend against at the victim end, currently the defense against these types of attacks lies on not getting compromised at the attacker end.

With denial of service attacks, a correct statement is

A They must use spoofed addresses since the attack stream must not complete the TCP three way hand shake to flood a network
B They are always based on filling a victims bandwidth up
C They can make use of flaws in the TCPIP stack and cause a machine to crash
D The sequence number of the ICMP packets is always the same in the flood traffic

ATTACK ANALYSES

Attack Tool Used

Wu-lnx.c is a program written in C. It was downloaded from www.securityfocus.com and compiled using gcc on a redhat 6.2 linux machine. This exploit code is intended to root compromise a target machine, which is running wu-ftpd. The exploit was run against a remote machine on the same segment running wu-ftpd 2.6.0 on redhat 6.2.

Correlations in the wild use of this style of attack

Within hours of the initial bugtraq posting on June 23 2000, scans for the exploit were detected and posted www.sans.org/y2k/062300-1430.htm
The CERT advisory CA-2000-13 describes this vulnerability.

5 months later scans for ftp continue top be prevalent on GIAC www.sans.org and in the CERT/CC Current Activity section http://www.cert.org/current/current_activity.html
This genre of attacks against wu-ftpd has been given CAN-2000-0574 (under review) by CVE. This type of attack is becoming more common as the idea has been published. Such vulnerabilities as the qpopper and rpc.statd are both examples of a format string attack.

Description of the attack

This is an input validate error in the site exec command within wu-ftpd. Since the printf command is given user input directly, it is possible to overwrite such things as a return address on the execution stack of the remote machine. The results are similar to the better-known buffer overflow attacks, but this attack is not an overflow, but a input validation error. If a section of code leaves out the format string command in the command, for instance sprintf, the format string can be passed to the command as input.

How to defend against this attack

The easiest after the fact method to defend is to upgrade to a new version of WU-FTPD from the appropriate vendor. Only accepting FTP connections from trusted partners using tcpwrappers is also a partial solution. If you must leave the service up unrestricted and cannot upgrade immediately, intensive logging of the port would be required. The CERT continues to receive examples of this vulnerability be used to exploit machines, http://www.cert.org/summaries/CS-2000-04.html

More work is being carried out on methods to prevent these coding problems from happening with code scanners that attempt to detect the common errors that lead to this family of attacks, see http://www.striker.ottawa.on.ca/~aland/pscan for an example of a code audit run against wu-ftpd.

Another defense mechanisms being evaluated is using compilers that attempt to disallow this type of validate problem from being exploited, see http://www.immunix.org/formatguard.html for an example of this method.

Annotated Traffic trace of the attack.
The attack was captured using Tcpdump and analyzed using Ethereal and SNORT.

Here is the ARP request when the exploit tries to make first contact with the victim machine

1 0.000000	00:00:00:00:00:00	00:50:da:ca:55:87	ARP	Who has 192.168.1.40? Tell 192.168.1.2
2 0.000233	00:50:da:ca:4c:71	00:00:00:00:00:01	ARP	192.168.1.40 is at 00:50:da:ca:4c:71

Three way hand shake to establish the FTP session

3 0.000254	192.168.1.2	192.168.1.40	TCP	1375 > ftp [SYN] Seq=3737694696 Ack=0 Win=32120 Len=0
4 0.000643	192.168.1.40	192.168.1.2	TCP	ftp > 1375 [SYN, ACK] Seq=3726417207 Ack=3737694697 Win=32120 Len=0

5 0.000698 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737694697 Ack=3726417208 Win=32120 Len=0

Banner showing the vulnerable version of FTP

6 3.015973 192.168.1.40 192.168.1.2 FTP Response: 220 Suffolk FTP server (Version wu-2.6.0(1) Sun Nov 12 14:42:19 EST 2000) ready.

Asking for a username and password (gets one, but also gets malicious shell code)

7 3.016049 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737694697 Ack=3726417290 Win=32120 Len=0
8 3.016966 192.168.1.2 192.168.1.40 FTP Request: user ftp

9 3.017194 192.168.1.40 192.168.1.2 TCP ftp > 1375 [ACK] Seq=3726417290 Ack=3737694706 Win=32120 Len=0
10 3.017226 192.168.1.2 192.168.1.40 FTP Request: pass

The anonymous or guest login is accepted (along with the malicious shell code)

11 3.020188 192.168.1.40 192.168.1.2 FTP Response: 331 Guest login ok, send your complete e-mail address as password.
12 3.020346 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695102 Ack=3726417358 Win=32120 Len=0
13 3.023202 192.168.1.40 192.168.1.2 FTP Response: 230 Guest login ok, access restrictions apply.
14 3.030346 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695102 Ack=3726417406 Win=32120 Len=0

The format strings are sent with the site exec command

15 4.021079 192.168.1.2 192.168.1.40 FTP Request: SITE EXEC %x %x %x %x +%x |%x

This continues

16 4.023245 192.168.1.40 192.168.1.2 FTP Response: 200-31 bffff24c 1ee 0 +bfffd1d4 |bfffcdd0
17 4.040344 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695132 Ack=3726417449 Win=32120 Len=0
18 4.040594 192.168.1.40 192.168.1.2 FTP Response: 200 (end of '%x %x %x %x +%x |%x')
19 4.060342 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695132 Ack=3726417486 Win=32120 Len=0
20 5.031221 192.168.1.2 192.168.1.40 FTP Request: SITE EXEC
21 5.033641 192.168.1.40 192.168.1.2 FTP Response: 200-31bffff24c1ee0bfffd1d4bfffcdd0bfffcdd4140140b
22 5.050343 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695324 Ack=3726417793 Win=31856 Len=0
23 5.050725 192.168.1.40 192.168.1.2 FTP Response: 200 (end of
24 5.070339 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695324 Ack=3726417992 Win=31657 Len=0
25 8.050563 192.168.1.2 192.168.1.40 FTP Request: SITE EXEC

26 8.054000 192.168.1.40 192.168.1.2 FTP Response: 200-aaaaaaaaaaaaaaaaaaaaaaaaaaabbbbxÍÿ¿-20-200-200
27 8.070351 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695809 Ack=3726418410 Win=31856 Len=0
28 8.070968 192.168.1.40 192.168.1.2 FTP Response:
29 8.090340 192.168.1.2 192.168.1.40 TCP 1375 > ftp [ACK] Seq=3737695809 Ack=3726418901 Win=31856 Len=0
30 9.061171 192.168.1.2 192.168.1.40 FTP Request: SITE EXEC

```
   31 9.074885   192.168.1.40      192.168.1.2       TCP    ftp > 1375 [ACK] Seq=3726418901 Ack=3737696307 Win=32120 Len=0
   32 19.847287  192.168.1.40      192.168.1.2       FTP    Response: 200-aaaaaaaaaaaaaaaaaaaaaaaaaaaaaabbbbxÍÿ¿-20-200
   33 19.848520  192.168.1.40      192.168.1.2       FTP    Response: 000000000000000000000000000000000000000000000000000
   34 19.849752  192.168.1.40      192.168.1.2       FTP    Response: 000000000000000000000000000000000000000000000000000
   35 19.849787  192.168.1.2       192.168.1.40      TCP    1375 > ftp [ACK] Seq=3737696307 Ack=3726422821 Win=31856 Len=0
   36 19.850982  192.168.1.40      192.168.1.2       FTP    Response: 000000000000000000000000000000000000000000000000000
   37 19.852311  192.168.1.40      192.168.1.2       FTP    Response: 000000000000000000000000000000000000000000000000000
   38 19.852338  192.168.1.2       192.168.1.40      TCP    1375 > ftp [ACK] Seq=3737696307 Ack=3726425717 Win=31856 Len=0
   39 19.853484  192.168.1.40      192.168.1.2       FTP    Response: 000000000000000000000000000000000000000000000000000
   40 19.870346  192.168.1.2       192.168.1.40      TCP    1375 > ftp [ACK] Seq=3737696307 Ack=3726427093 Win=31856 Len=0
   41 19.870584  192.168.1.40      192.168.1.2       FTP    Response:
000000000000000000000000000000000000000000000000000
   42 19.890345  192.168.1.2       192.168.1.40      TCP    1375 > ftp [ACK] Seq=3737696307 Ack=3726427094 Win=31856 Len=0
   43 39.014844  192.168.1.2       192.168.1.40      TCP    1375 > ftp [FIN, ACK] Seq=3737696307 Ack=3726427094 Win=31856 Len=0
   44 39.015092  192.168.1.40      192.168.1.2       TCP    ftp > 1375 [ACK] Seq=3726427094 Ack=3737696308 Win=32120 Len=0
   45 39.016783  192.168.1.40      192.168.1.2       TCP    ftp > 1375 [FIN, ACK] Seq=3726427094 Ack=3737696308 Win=32120 Len=0
   46 39.016823  192.168.1.2       192.168.1.40      TCP    1375 > ftp [ACK] Seq=3737696308 Ack=3726427095 Win=31856 Len=0
```

Key portions of the above trace with the payload included:

Frame 10 payload shows the format strings being sent in the username/password combination

```
Frame 10 (462 on wire, 462 captured)
   Arrival Time: Nov 13, 2000 10:11:00.9644
   Time delta from previous packet: 0.000032 seconds
   Frame Number: 10
   Packet Length: 462 bytes
   Capture Length: 462 bytes
Ethernet II
   Destination: 00:50:da:ca:55:87 (00:50:da:ca:55:87)
   Source: 00:00:00:00:00:00 (00:00:00:00:00:00)
   Type: IP (0x0800)
Internet Protocol
   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..00 = Currently Unused: 0
```

Total Length: 448
Identification: 0x850d
Flags: 0x04
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0x30b0 (correct)
Source: 192.168.1.2 (192.168.1.2)
Destination: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: 1375 (1375), Dst Port: ftp (21), Seq: 3737694706, Ack: 3726417290
Source port: 1375 (1375)
Destination port: ftp (21)
Sequence number: 3737694706
Acknowledgement number: 3726417290
Header length: 32 bytes
Flags: 0x0018 (PSH, ACK)
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 1... = Push: Set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 32120
Checksum: 0xbc44
Options: (12 bytes)
NOP
NOP
Time stamp: tsval 100027894, tsecr 17339487
File Transfer Protocol
Request: pass
Request Arg:

Frame 11 shows no error or abnormal end based on the unexpected input passed to ftpd

Frame 11 (134 on wire, 134 captured)
    Arrival Time: Nov 13, 2000 10:11:00.9674
    Time delta from previous packet: 0.002962 seconds
    Frame Number: 11
    Packet Length: 134 bytes
    Capture Length: 134 bytes
Ethernet II
    Destination: 00:00:00:00:00:01 (00:00:00:00:00:01)
    Source: 00:50:da:ca:4c:71 (00:50:da:ca:4c:71)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP)
        0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
        .... ..00 = Currently Unused: 0
    Total Length: 120
    Identification: 0x0658
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xb09d (correct)
    Source: 192.168.1.40 (192.168.1.40)
    Destination: 192.168.1.2 (192.168.1.2)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: 1375 (1375), Seq: 3726417290, Ack: 3737695102
    Source port: ftp (21)
    Destination port: 1375 (1375)
    Sequence number: 3726417290
    Acknowledgement number: 3737695102
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set

.... 1... = Push: Set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 32120
Checksum: 0x8f22
Options: (12 bytes)
NOP
NOP
Time stamp: tsval 17339487, tsecr 100027894
File Transfer Protocol
Response: 331
Response Arg: Guest login ok, send your complete e-mail address as password.

Frame 20 shows the SITE EXEC command followed by the format string characters

Frame 20 (258 on wire, 258 captured)
Arrival Time: Nov 13, 2000 10:11:02.9784
Time delta from previous packet: 0.970879 seconds
Frame Number: 20
Packet Length: 258 bytes
Capture Length: 258 bytes
Ethernet II
Destination: 00:50:da:ca:55:87 (00:50:da:ca:55:87)
Source: 00:00:00:00:00:00 (00:00:00:00:00:00)
Type: IP (0x0800)
Internet Protocol
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..00 = Currently Unused: 0
Total Length: 244
Identification: 0x8513
Flags: 0x04
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0

Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0x3176 (correct)
Source: 192.168.1.2 (192.168.1.2)
Destination: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: 1375 (1375), Dst Port: ftp (21), Seq: 3737695132, Ack: 3726417486
Source port: 1375 (1375)
Destination port: ftp (21)
Sequence number: 3737695132
Acknowledgement number: 3726417486
Header length: 32 bytes
Flags: 0x0018 (PSH, ACK)
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 1... = Push: Set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 32120
Checksum: 0x180b
Options: (12 bytes)
NOP
NOP
Time stamp: tsval 100028096, tsecr 17339589
File Transfer Protocol
Request: SITE
Request Arg: EXEC
%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x
%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x|%
x

Frame 25 shows more unexpected input being injected through SITE EXEC

Frame 25 (551 on wire, 551 captured)
Arrival Time: Nov 13, 2000 10:11:05.9977
Time delta from previous packet: 2.980224 seconds
Frame Number: 25

```
    Packet Length: 551 bytes
    Capture Length: 551 bytes
Ethernet II
    Destination: 00:50:da:ca:55:87 (00:50:da:ca:55:87)
    Source: 00:00:00:00:00:00 (00:00:00:00:00:00)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Currently Unused: 0
    Total Length: 537
    Identification: 0x8516
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x304e (correct)
    Source: 192.168.1.2 (192.168.1.2)
    Destination: 192.168.1.40 (192.168.1.40)
Transmission Control Protocol, Src Port: 1375 (1375), Dst Port: ftp (21), Seq: 3737695324, Ack: 3726417992
    Source port: 1375 (1375)
    Destination port: ftp (21)
    Sequence number: 3737695324
    Acknowledgement number: 3726417992
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 31856
```

Checksum: 0x02fe
Options: (12 bytes)
NOP
NOP
Time stamp: tsval 100028398, tsecr 17339690
File Transfer Protocol
Request: SITE
Request Arg: EXEC
aaaaaaaaaaaaaaaaaaaaaaaaabbbbxÍÿÿ¿%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f

Snort Alerts Generated by the attack

Alert IDS181 triggers on the x86 Shell Code that is injected by the attack into the victim.

[**] IDS181 - MISC - Shellcode X86 NOPS [**]
11/13-10:11:00.964452 192.168.1.2:1375 -> 192.168.1.40:21
TCP TTL:64 TOS:0x0 ID:34061 DF
AP Seq: 0xDEC8B1F2 Ack: 0xDE1C9D8A Win: 0x7D78
TCP Options => NOP NOP TS: 100027894 17339487
Alert IDS317 shows the issuing of the SITE EXEC command
[**] IDS317 - FTP-site-exec [**]
11/13-10:11:01.968305 192.168.1.2:1375 -> 192.168.1.40:21
TCP TTL:64 TOS:0x0 ID:34064 DF
AP Seq: 0xDEC8B37E Ack: 0xDE1C9DFE Win: 0x7D78
TCP Options => NOP NOP TS: 100027995 17339487

Alert IDS317 Repeats during the attack.

ANALYZE THIS

We are pleased to submit this summary of the findings of your submitted network logs as part of our comprehensive proposal for your network security needs. Where our team provided specific advice for blocking intruders, we assumed routing equipment from CISCO and firewalls based on IPCHAINS. Cisco was chosen given its market share, and IPCHAINS was chosen based on the fact you already use an open source IDS system Snort.

Some Caveats

Two Way Communications

In many cases we only have 1/2 of the communications sequence. Without the full capture of the communications, we are forced to make suppositions about what might have preceded the scan or alert.

Time Gaps

We noted that there are a number of days that are unaccounted for. Please note this when considering the statistics and summaries we provide in particular. This should not affect the overall analysis of the data.

False positives

These results are best evaluated in a meeting between your network staff and our security staff, as your staff brings the network context to help weed out the false positives. Your own network administrators are in the best position to confirm these detects as legitimate.

The corollary of this is False Negatives, which would mean items of interest that the IDS was not capable of or configured to catch. This should be kept in mind when reviewing the data.

Security Posture

Overall the fact that you employ an IDS system on your networks puts you in the top stratum of networks that we are called into consult on, your staff should be commended. We still believe our team could provide an excellent value added to help your staff deal with what is a very active network.

One of they key aspects of your analysis of this report is for you to understand what makes up a legitimate threat to your network. To better help you understand the data in the support, the most significant detects have been correlated to other sources to help build a complete picture.

The client provided watch list was also given special attention. Without knowing the exact reason these sources are on a watch list, the results must be interpreted by your analytical staff to be put into proper context so you can build a threat assessment.

Virus Activity

While there was evidence of virus activity, the team assumed that appropriate countermeasures should be placed at the mail server with virus scanning software. While network countermeasures can defend against certain viruses, the best solution in our experience is a SMTP based mail scanner. Out team would be happy to provide advice and guidance on this, however it is outside the scope of this proposal.

The two IP addresses that received the HAPPY99 virus are.

08/16-14:36:46.954418 [**] Happy 99 Virus [**] 128.8.198.101:12805 -> MY.NET.6.35:25
08/20-15:41:12.157972 [**] Happy 99 Virus [**] 24.2.2.66:58102 -> MY.NET.179.80:25

Our Methodology

The team faced a challenge on analyzing over 480000 lines of data with out some contextual information about the network to help prioritize the work. Also we are not aware of the security posture or what threat risk assessment your company might have.

The team made use of perl scripts from www.sans.org/y2k/practical/lenny_zeltser.htm www.sans.org/y2k/practical/bill_royds.zip. Other techniques used for extracting the data included Microsoft Excel and common Unix parsing commands such as grep, diff and sort. The commands used to extract the data are included so you may verify our results with your data.

The process we follow when doing an audit of any data set we are unfamiliar with is somewhat manual in nature, after we become more familiar with the context your data is taken from, we then attempt to automate many of the processes. While this in part manual process is more time consuming, it lets the analysts get a better feel for your network as the view the data. The second portion of our analyses was based on looking through your data for what the Internet as a whole is seeing in terms of threat activity. This picture is provided from such reputable sources as the SANS TOP 10 at www.sans.org and the http://www.cert.org/current/current_activity.html. The team also used an assumption of noting one connection to a port as a scan, 2-3 connections as a repeated scan. If one source IP connects to one destination IP, on the same port more than 4 times, we reported it as a possible access to that port.

Some port numbers are used for more than one program. What is detected as a Trojan scans to a high port could be normal high port communications of a legitimate session. With out more surrounding details of your data, the team tended to err on the side of caution and flag the event.

When reviewing the data keep in mind your knowledge of the network, for instance the following logs:

Aug 15 02:08:59 195.114.226.41:4387 -> MY.NET.146.88:21 SYN **S*****
Aug 15 02:09:08 195.114.226.41:4387 -> MY.NET.146.88:21 SYN **S*****
Aug 15 02:13:10 195.114.226.41:4387 -> MY.NET.161.235:21 SYN **S*****
Aug 15 02:24:50 195.114.226.41:4387 -> MY.NET.208.180:21 SYN **S*****
Aug 15 02:24:53 195.114.226.41:4387 -> MY.NET.208.180:21 SYN **S*****

Are part of a very large scan for ftp on your network. If you did not offer this service on your network, then the risk is reduced for you.

Not all scans are readily explainable. This log segment

Aug 28 15:39:03 198.62.155.105:41365 -> MY.NET.217.10:515 SYN **S*****
Aug 28 15:40:46 198.62.155.106:42626 -> MY.NET.217.10:515 SYN **S*****
Aug 28 15:32:31 207.236.3.96:3480 -> MY.NET.20.10:515 SYN **S*****

Sep 4 11:42:16 216.99.200.242:17421 -> MY.NET.97.216:515 SYN **S*****
Sep 13 16:52:45 216.99.200.242:18544 -> MY.NET.98.188:515 SYN **S*****
Sep 11 04:51:42 24.180.134.156:1227 -> MY.NET.208.37:515 SYN **S*****

Indicated scanning for port 515. Not until November 21 did the community come to a conclusion on what this was:

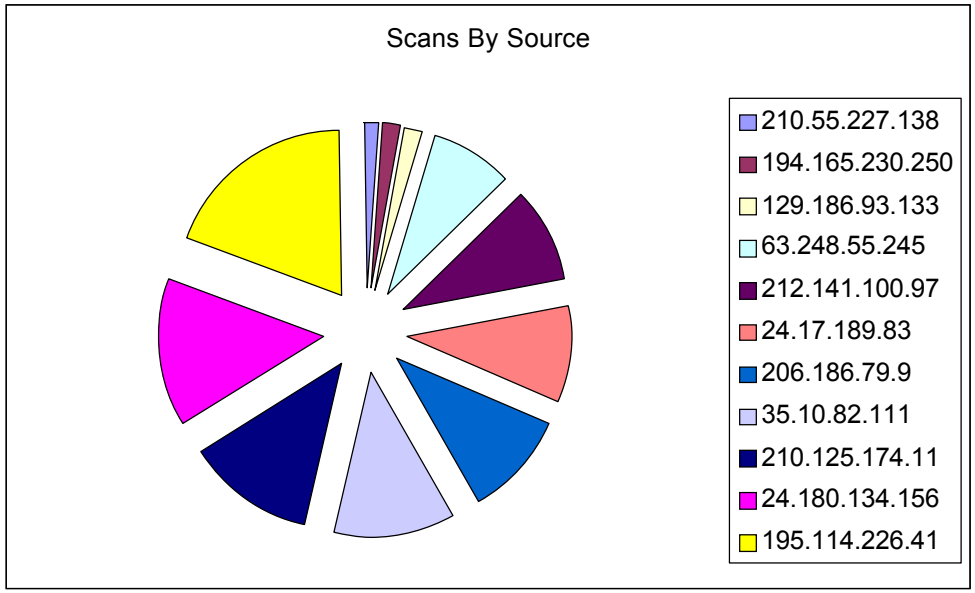
http://www.sans.org/newlook/alerts/port515.htm

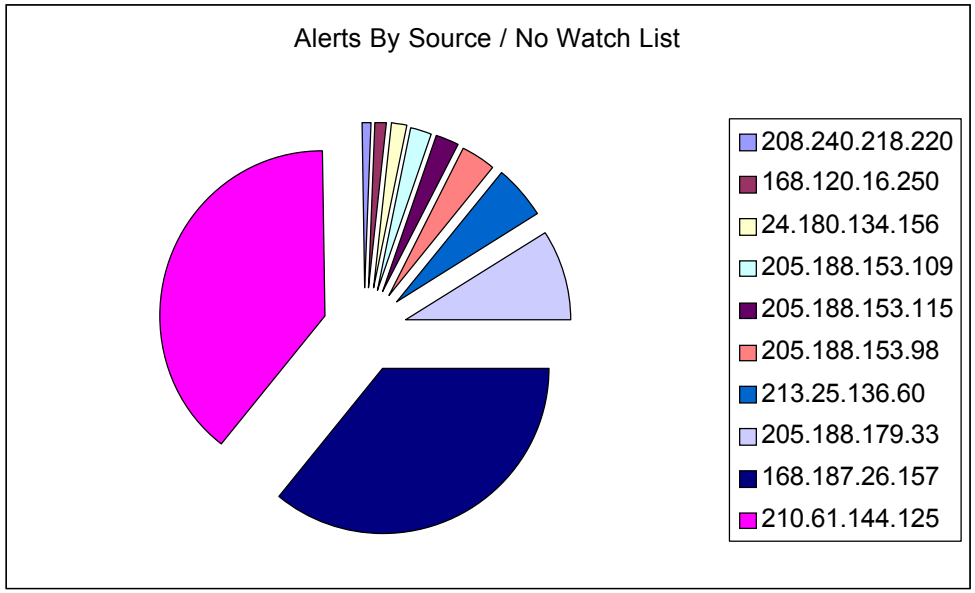
The LPRng port, versions prior to 3.6.24, contains a potential vulnerability which may allow root compromise from both local and remote systems. The vulnerability is due to incorrect usage of the syslog(3) function. Local and remote users can send string-formatting operators to the printer daemon to corrupt the daemon's execution, potentially gaining root access.

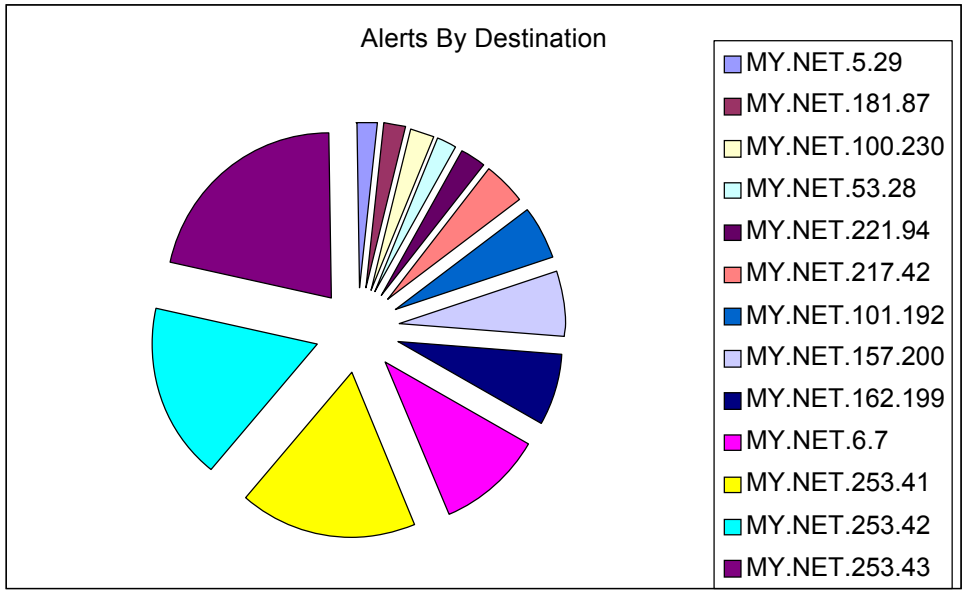
SIGIFIGANCE OF THE BELOW CHARTS

What is important to note about these charts is the large volume of detects. There is a definite threat against your network. However, keep in mind that many IDS alerts can be false positive, so the numbers can produce an image of a network under siege and perhaps you might consider surrendering, we do not think it is that bad.

The team also separated out the watch list in some charts, the view of the threat changes once this is done. As the watchlist has already been identified as a threat, the second view might be more interesting.







REPORT DETAILS

Section 1 - Overview of top threats detected on your network
Section 2 - Summary of threat to Core servers
Section 3 - Summary of threat from Top Source IP
Section 4 - Summary of threat to Top Target IP
Section 5 - Summary of threat from your provided watchlist
Section 6 – Miscellaneous
Section 7 - Recommendations

SECTION 1

Some Significant Detects

SNMP Public Access

SMB Name Wildcards

Various machines are accessing MY.NET.101.192 through SNMP Public Access and SMB Name Wildcards. This poses a risk for a number of reasons.

09/11-18:31:00.333609 [**] SNMP public access [**] MY.NET.97.217:1066 -> MY.NET.101.192:161
09/03-12:42:35.351628 [**] SNMP public access [**] MY.NET.98.109:1045 -> MY.NET.101.192:161
09/10-15:27:45.937458 [**] SNMP public access [**] MY.NET.98.172:1042 -> MY.NET.101.192:161

Simple Network Management Protocol, is a mechanism for controlling and getting information from network devices. Community strings are the SNMP version of authentication. The default community strings are PUBLIC for read and PRIVATE for write. Read access gives you the ability to get information from the network node, such as uptime, network traffic stats etc. Write access gives you the ability affect changes on the device, such as disabling a port. The default SNMP password is number 10 on the SANS Top Internet Threat consensus

CVE Entries:
Default or blank SNMP community name (public) - CAN-1999-0517
Guessable SNMP community name - CAN-1999-0516
Hidden SNMP community strings - CAN-1999-0254, CAN-1999-0186

Additional Information can be found at
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid210315

There are other possibilities on what was being scanned for, such as www.securityfocus.com BUGTRAQ ID 177 and 1177, which describe SNMP software with vulnerabilities beyond the more common default password problems.

NETBIOS scans are the Sans Top 10 # 7and the following CERT info IN-2000-02 IN-2000-03

08/15-20:00:00.271636 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
08/15-20:00:00.271636 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137

NETBIOS is a common component of Microsoft Networking. Sharing network resources through drive mappings is a common feature implemented over netbios. Having wildcard access available over the Internet poses a risk. The SMB wildcard indicates that Microsoft networking is taking place. This type of networking internally might be an acceptable risk as in the above sample, however

08/11-16:42:10.039852 [**] SMB Name Wildcard [**] 206.171.108.1:724 -> MY.NET.6.7:137
08/11-16:42:11.495774 [**] SMB Name Wildcard [**] 206.171.108.1:724 -> MY.NET.6.7:137

Shows this possibly taking place from the outside. This is a more serious risk if the .6.7 machine does offer netbios to the Internet.

Possible Open Proxies
CERT CA-98.03

A proxy is a way to redirect your Internet traffic through a middle machine, known as a proxy server. In the case of an open proxy server, IP addresses from outside your network are continually connected to the proxy server port. This could indicate that you are offering a way to anonymize traffic for others through your network. This one is prioritized since it on a numerically low IP.

09/11-18:44:38.004677 [**] WinGate 1080 Attempt [**] 168.187.26.157:3702 -> MY.NET.6.7:1080
Repeats Multiple Times

Inbound Telnet activity from Watch List

08/16-20:43:29.383282 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:4628 -> MY.NET.6.7:23
08/16-20:43:30.424659 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:4628 -> MY.NET.6.7:23
08/16-20:43:32.491068 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:4628 -> MY.NET.6.7:23
Repeats Multiple Times

This could indicate a telnet session from your watch list IP to your server. Or an attempt at password guessing. The fact that the source port does not change could indicate a type of tool is being used. CERT IN-2000-09 describes some various telnet issues especially if you are running IRIX server.

RPC repeated access from Italian ISP

09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:40.010708 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:40.010708 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:40.010708 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:40.010708 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771

Repeats Multiple Times from multiple sources with multiple alarm types.

08/18-06:10:13.626765 [**] External RPC call [**] 18.116.0.75:1661 -> MY.NET.6.15:111
08/18-06:10:13.626765 [**] External RPC call [**] 18.116.0.75:1661 -> MY.NET.6.15:111
08/18-06:10:13.626765 [**] External RPC call [**] 18.116.0.75:1661 -> MY.NET.6.15:111
08/18-12:13:28.981943 [**] SMB Name Wildcard [**] 4.17.88.66:137 -> MY.NET.6.15:137
08/18-12:13:28.981943 [**] SMB Name Wildcard [**] 4.17.88.66:137 -> MY.NET.6.15:137
08/19-01:39:20.501009 [**] External RPC call [**] 141.223.124.31:2796 -> MY.NET.6.15:111
08/19-01:39:20.501009 [**] External RPC call [**] 141.223.124.31:2796 -> MY.NET.6.15:111
08/19-01:46:08.843875 [**] External RPC call [**] 141.223.124.31:875 -> MY.NET.6.15:111
08/19-01:46:08.843875 [**] External RPC call [**] 141.223.124.31:875 -> MY.NET.6.15:111
08/19-10:11:34.529565 [**] External RPC call [**] 209.160.238.215:2572 -> MY.NET.6.15:111
08/19-10:11:34.529565 [**] External RPC call [**] 209.160.238.215:2572 -> MY.NET.6.15:111
08/19-10:11:34.529565 [**] External RPC call [**] 209.160.238.215:2572 -> MY.NET.6.15:111
08/19-12:12:52.958110 [**] External RPC call [**] 209.160.238.215:782 -> MY.NET.6.15:111
09/02-00:28:03.467564 [**] SYN-FIN scan! [**] 210.101.101.110:111 -> MY.NET.6.15:111
09/02-00:28:03.467564 [**] SYN-FIN scan! [**] 210.101.101.110:111 -> MY.NET.6.15:111
09/02-00:28:03.467564 [**] SYN-FIN scan! [**] 210.101.101.110:111 -> MY.NET.6.15:111
09/02-00:28:03.467564 [**] SYN-FIN scan! [**] 210.101.101.110:111 -> MY.NET.6.15:111
09/02-00:28:06.989407 [**] External RPC call [**] 210.101.101.110:861 -> MY.NET.6.15:111
09/02-00:28:06.989407 [**] External RPC call [**] 210.101.101.110:861 -> MY.NET.6.15:111
09/02-00:28:06.989407 [**] External RPC call [**] 210.101.101.110:861 -> MY.NET.6.15:111
09/02-00:28:06.989407 [**] External RPC call [**] 210.101.101.110:861 -> MY.NET.6.15:111
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/02-09:39:11.608534 [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/03-11:42:36.543439 [**] External RPC call [**] 210.100.199.219:2378 -> MY.NET.6.15:111
09/03-11:42:36.543439 [**] External RPC call [**] 210.100.199.219:2378 -> MY.NET.6.15:111
09/03-11:42:36.543439 [**] External RPC call [**] 210.100.199.219:2378 -> MY.NET.6.15:111
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/07-05:37:39.663140 [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
09/10-03:15:34.241029 [**] External RPC call [**] 161.31.208.237:875 -> MY.NET.6.15:111
09/10-03:15:34.241029 [**] External RPC call [**] 161.31.208.237:875 -> MY.NET.6.15:111
09/10-03:15:34.249462 [**] External RPC call [**] 161.31.208.237:2223 -> MY.NET.6.15:111
09/10-03:15:34.249462 [**] External RPC call [**] 161.31.208.237:2223 -> MY.NET.6.15:111

If this machine does not offer external RPC style services, it should be observed to see if there was any compromise as activity to the RPC (ruserd commonly) service could be exploited. This type of scanning for 111 the portmapper and the various high ports that the RPC services bind to is very common due to the variety of RPC exploits that have appeared. The following is an example of other sites being scanned for the same thing.

http://www.sans.org/y2k/101600-1130.htm

Oct 12 08:16:40 hostj snort[24697]: MISC-Attempted Sun RPC high port access: 208.19.16.150:930 -> z.y.w.66:32771
Oct 12 08:16:41 hostj snort[24697]: IDS181 - MISC - Shellcode X86 NOPS: 208.19.16.150:930 -> z.y.w.66:32771
Oct 12 08:16:41 hostj snort[24697]: MISC-Attempted Sun RPC high port access: 208.19.16.150:930 -> z.y.w.66:32771

RPC Scans are the SANS Top 10 # 3 and have the following CERT info IN-2000-10 CA-2000-17 CA-99-16.

Section 2

Core Server Activity

Our analytical team made certain assumptions based on our brief overview of your data, these assumptions are listed as follows:

The following servers appear to be core servers based on their apparent legitimate DNS or E-MAIL traffic. Since these servers are running critical services, they were given more analytical attention by the team. Also many organizations use a convention of keeping infrastructure machines numerically lower in the IP ranger. For this reason detects against any servers numbered less than .10 were given a high degree of importance. We realize these assumptions might not hold true for your network, but even without the assumptions, we believe the analysis should hold true, it would be the severity of the incidents that would be mainly affected.

Low IP Ranges MY.NET.1.X
Low IP Ranges MY.NET.X.[1-10]

Incoming Mail

09/08-08:15:37.459156 [**] Watchlist 000222 NET-NCFC [**] 159.226.228.1:4965 -> MY.NET.110.150:25
09/08-08:15:37.459156 [**] Watchlist 000222 NET-NCFC [**] 159.226.228.1:4965 -> MY.NET.110.150:25
This Repeats.

The amount of traffic in this above log indicates that mail transfer is taking place. These IP addresses appear to be mail servers, but are possibly receiving mail from a monitored IP on your watch list.

DNS Servers

The amount of traffic seems to indicate there are DNS clients pointing outside your network.

Sep 10 01:57:53 136.160.7.2:53 -> MY.NET.115.115:1112 UDP
Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1150 UDP

MY.NET.1.3 and MY.NET.1.4 and MY.NET.1.5 appear to offer DNS and NTP. NTP is normally associated with core servers or routers on a network.

Sep 3 09:05:47 MY.NET.1.5:123 -> MY.NET.100.111:123 UDP
Sep 3 09:05:54 MY.NET.1.5:123 -> MY.NET.100.121:123 UDP
Sep 3 09:09:18 MY.NET.1.4:123 -> MY.NET.100.169:123 UDP
Sep 3 09:05:35 MY.NET.1.3:123 -> MY.NET.99.66:123 UDP

MY.NET.1.13 seem to offer a service of Andrew File System based on the port 7000 traffic. Our team is unfamiliar with AFS, but we did note the following.

MY.NET.1.13 Connects out a large number of your internal MY.NET.systems on port 7001 and 7002.
MY.NET.1.13 Connects out a large number of your internal MY.NET.systems on port 111.

Sep 3 09:09:06 MY.NET.1.13:7003 -> MY.NET.60.165:7001 UDP
Sep 3 09:09:10 MY.NET.1.13:7003 -> MY.NET.60.165:7001 UDP
Sep 3 09:03:21 MY.NET.1.13:7003 -> MY.NET.60.167:7001 UDP
Sep 3 09:03:23 MY.NET.1.13:7003 -> MY.NET.60.167:7001 UDP
Sep 3 09:09:22 MY.NET.1.13:7003 -> MY.NET.60.168:7001 UDP
Sep 3 09:09:29 MY.NET.1.13:7003 -> MY.NET.60.168:7001 UDP

More information on AFS can be found at http://www.sans.org/y2k/practical/milind_saraph.html

The team also noted DNS-DNS (53udp-53udp) traffic with an outside organization that appeared in your scan records.

Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.110.100:53 UDP
Aug 28 01:58:11 136.160.7.2:53 -> MY.NET.110.131:53 UDP
Aug 28 01:58:15 136.160.7.2:53 -> MY.NET.110.131:53 UDP
Sep 9 01:57:53 136.160.7.2:53 -> MY.NET.110.131:53 UDP
Sep 10 01:57:53 136.160.7.2:53 -> MY.NET.115.115:1112 UDP
Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1150 UDP
Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1158 UDP
Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1168 UDP
Sep 10 01:57:58 136.160.7.2:53 -> MY.NET.115.115:1170 UDP

Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1175 UDP
Sep 10 01:57:54 136.160.7.2:53 -> MY.NET.115.115:1177 UDP
Sep 10 01:57:55 136.160.7.2:53 -> MY.NET.115.115:1181 UDP
Sep 10 01:57:58 136.160.7.2:53 -> MY.NET.115.115:1196 UDP
Sep 10 01:57:58 136.160.7.2:53 -> MY.NET.115.115:1197 UDP

It would appear 131.160.7.2 (USMD) is using your DNS server. If they are a legitimate business partner, this might be normal.

ALERTS TO CORE SERVERS

A number of scans target or pass by your core server range MY.NET.1.X, here are a few examples and their significance. IP ranges that have been pre flagged by your watchlist that are targeting the core range warrant special attention based on your knowledge of the watchlist rational.

09/14-09:15:35.433598 [**] Tiny Fragments - Possible Hostile Activity [**] 62.76.42.18 -> MY.NET.1.8
09/14-09:15:35.433598 [**] Tiny Fragments - Possible Hostile Activity [**] 62.76.42.18 -> MY.NET.1.8
09/14-09:15:35.939752 [**] Tiny Fragments - Possible Hostile Activity [**] 62.76.42.18 -> MY.NET.1.9
09/14-09:15:44.375639 [**] Tiny Fragments - Possible Hostile Activity [**] 62.76.42.18 -> MY.NET.1.10

Separating the TCP/IP networking packets into very small fragments is normally an attempt to cause a system to crash, or to gain information or access a system without being detected by an IDS. Fragmentation on a network is a normal occurrence; fragmentation that is small enough to trip the IDS typically is cause for concern. Also of note is the targeting of numeric low addresses were core servers typically live and that the only activity generated by the IP was the fragmented traffic. The host resolves to a Russian Educational Institute.

08/18-02:21:02.892869 [**] NMAP TCP ping! [**] 205.128.11.157:53 -> MY.NET.1.8:53
08/18-02:21:02.892958 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/18-02:21:02.892958 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/18-19:07:59.701158 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/18-19:07:59.701158 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
08/20-08:42:27.214025 [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53

09/03-04:34:51.432240 [**] NMAP TCP ping! [**] 202.187.24.3:80 -> MY.NET.60.14:80
09/03-04:34:51.432240 [**] NMAP TCP ping! [**] 202.187.24.3:80 -> MY.NET.60.14:80
09/07-03:52:20.386042 [**] NMAP TCP ping! [**] 202.187.24.3:80 -> MY.NET.179.77:80
09/12-22:36:20.070878 [**] NMAP TCP ping! [**] 202.187.24.3:80 -> MY.NET.1.3:53

08/15-14:11:15.897081 [**] NMAP TCP ping! [**] 209.218.228.201:53 -> MY.NET.1.8:53
08/15-14:11:15.897081 [**] NMAP TCP ping! [**] 209.218.228.201:53 -> MY.NET.1.8:53

NAMP is a popular network-scanning tool. It issues a NMAP ping before scanning hosts to see if they are up. A wide variety of information about a host, such as what services it offers and what operating system it runs can be discovered by a full NAMP scan. This can be a preamble to a more serious attack one the adversary knows what services you offer, and what OS you run, they can properly target their exploits. Again these target core servers.

Aug 18 02:44:26 4.54.37.160:3825 -> MY.NET.6.203:21 SYN **S*****
Aug 18 02:44:38 4.54.37.160:3825 -> MY.NET.6.203:21 SYN **S*****
Aug 18 02:46:46 4.54.37.160:4014 -> MY.NET.60.213:21 SYN **S*****
Aug 18 02:46:45 4.54.37.160:4016 -> MY.NET.60.215:21 SYN **S*****
Aug 18 02:48:02 4.54.37.160:4038 -> MY.NET.1.3:21 SYN **S*****
Aug 18 02:48:02 4.54.37.160:4040 -> MY.NET.1.5:21 SYN **S*****

4.54.37.160 scanned for FTP using a SYN only scan. A SYN only scan is an attempt to evade logging devices by opening the SYN of a TCP three way hand shake and then not responding to the ACK. The attacker targeted 3 subnets, one of which is the MY.NET.1.1, a core subnet

SECTION 3

Some Top Source IP Scans and Alerts

09/11-18:40:36.435240 [**] WinGate 1080 Attempt [**] 168.187.26.157:1518 -> MY.NET.1.9:1080
Repeats for entire range

08/11-00:46:56.976538 [**] WinGate 1080 Attempt [**] 208.240.218.220:4390 -> MY.NET.217.46:1080
Repeats for large range

Syn-fin scans are an attempt to defeat IDS and other logging mechanisms. With today's modern IDS system they almost work in reverse sometimes standing out more than a regular connection would. Nonetheless, they are an attempt at creating a packet that will scan and evade detection. The NULL scan further down the logs is another attempt at evading detection or fingerprinting the stack by evaluating responses to odd packets.

09/11-06:45:13.077482 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.1.3:21
Repeats for large range

The sequence number of this packet does not change as would be expected, which would add to the suspicious nature of this packet. While three packets might have the same number if TCP is retrying, the not repeating sequence continues on.

=+
09/11-06:45:15.232761 210.61.144.125:21 -> MY.NET.1.22:21

```
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x2EA8E085   Ack: 0x40C83D4C   Win: 0x404
00 00 00 00 00 00                           ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/11-06:45:15.354553 210.61.144.125:21 -> MY.NET.1.28:21
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x2EA8E085   Ack: 0x40C83D4C   Win: 0x404
00 00 00 00 00 00                           ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/11-06:45:15.392526 210.61.144.125:21 -> MY.NET.1.30:21
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x2EA8E085   Ack: 0x40C83D4C   Win: 0x404
00 00 00 00 00 00                           ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Other Internet Dwellers are seeing the same type of scan as indicated by

```
Sep 21 17:27:21 hostp in.ftpd[23546]: connect from 210.61.144.125
Sep 21 17:27:21 hostp in.ftpd[23547]: connect from 210.61.144.125
Sep 21 17:28:40 hostca in.ftpd[17331]: connect from 210.61.144.125
Sep 21 17:28:40 hostca in.ftpd[17333]: connect from 210.61.144.125
Sep 21 17:28:43 hostba in.ftpd[2779]: refused connect from 210.61.144.125
Sep 21 17:30:32 hostmau Connection attempt to
  TCP 198.82.161.28:21 from 210.61.144.125:21
Sep 21 17:36:02 hosty snort[395880]: SCAN-SYN FIN:
  210.61.144.125:21 >z.y.w.34:21
Sep 21 17:36:03 hostj snort[341]: SCAN-SYN FIN:
  210.61.144.125:21 -> z.y.w.66:21
Sep 21 17:36:03 hostmi snort[15718]: SCAN-SYN FIN:
  210.61.144.125:21 >z.y.w.98:21
```

The source of this IP is:

Organization Name First Securities Co., LTD.

Street Address 12F, No. 39, Sec. 2, Tun-Hwa S. Rd,
City Taipei
State Taiwan
Country Code TW
IP Network 210.61.144.64/26
Network Name FSCL-NET

09/07-21:33:23.187413 [**] SYN-FIN scan! [**] 213.25.136.60:9704 -> MY.NET.1.4:9704
Repeats for large range

24.180.134.156 generates a significant amount of traffic scanning for proxies and fingerprinting machines. Machines from the 24.X.X.X network are typically cable modem users. Some of our other clients have found paying special attention to this block of addresses is warranted to due to many scans from this block.

09/11-04:48:08.283345 [**] WinGate 1080 Attempt [**] 24.180.134.156:3931 -> MY.NET.208.1:1080
09/11-04:48:14.514481 [**] NMAP TCP ping! [**] 24.180.134.156:50114 -> MY.NET.208.1:35829
09/11-04:48:32.719690 [**] NMAP TCP ping! [**] 24.180.134.156:50114 -> MY.NET.208.2:39069
09/11-04:48:42.827361 [**] NMAP TCP ping! [**] 24.180.134.156:50114 -> MY.NET.208.2:40026
09/11-04:48:56.731170 [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.5:23
09/11-04:48:56.737089 [**] NMAP TCP ping! [**] 24.180.134.156:50112 -> MY.NET.208.5:23
09/11-04:48:56.744829 [**] NMAP TCP ping! [**] 24.180.134.156:50114 -> MY.NET.208.5:34552
09/11-04:49:18.298916 [**] WinGate 1080 Attempt [**] 24.180.134.156:2330 -> MY.NET.208.9:1080
09/11-04:49:39.789570 [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.13:23
09/11-04:49:52.852370 [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.17:23
09/11-04:49:52.855709 [**] NMAP TCP ping! [**] 24.180.134.156:50112 -> MY.NET.208.17:23
09/11-04:50:26.892549 [**] WinGate 1080 Attempt [**] 24.180.134.156:4645 -> MY.NET.208.21:1080
09/11-04:50:32.156163 [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.21:23
09/11-04:50:32.160549 [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.21:23
09/11-04:50:45.075709 [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.25:23
09/11-04:50:45.077348 [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.25:23
Repeats for large range

More proxy scanning

09/02-00:20:56.463518 [**] WinGate 1080 Attempt [**] 168.120.16.250:55419 -> MY.NET.97.212:1080
Repeats for large range

Some FTP scans likely looking for recent wu-ftpd exploits see CERT entries CA-2000-13 AA-2000.02 CA-99-13

09/11-06:45:21.805565 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.2.184:21
09/11-06:45:21.805565 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.2.184:21
09/11-06:45:21.805565 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.2.184:21
09/11-06:45:21.805565 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.2.184:21

And some telnet scans See CERT IN-2000-09

Sep 6 12:58:22 128.171.57.194:1522 -> MY.NET.18.120:23 SYN **S*****
Sep 6 12:58:22 128.171.57.194:1523 -> MY.NET.18.121:23 SYN **S*****
Sep 6 12:58:22 128.171.57.194:1526 -> MY.NET.18.124:23 SYN **S*****

Some scans for DNS which is the SANS Top 10 # 1 CERT CA-2000-02 IN-2000-04 CA-2000-03 CA-99-14

Sep 10 01:56:22 206.186.79.9:4257 -> MY.NET.210.231:53 SYN **S*****
Sep 10 01:14:22 206.186.79.9:4293 -> MY.NET.109.209:53 SYN **S*****
Sep 10 01:14:22 206.186.79.9:4294 -> MY.NET.109.210:53 SYN **S*****
Sep 10 01:14:22 206.186.79.9:4295 -> MY.NET.109.211:53 SYN **S*****
Sep 10 01:14:22 206.186.79.9:4296 -> MY.NET.109.212:53 SYN **S*****
Sep 10 01:14:22 206.186.79.9:4301 -> MY.NET.109.217:53 SYN **S*****
Sep 10 00:04:22 206.186.79.9:4341 -> MY.NET.209.223:53 SYN **S*****
Sep 10 00:04:22 206.186.79.9:4342 -> MY.NET.209.224:53 SYN **S*****
Sep 10 00:04:22 206.186.79.9:4348 -> MY.NET.209.230:53 SYN **S*****
Sep 10 00:04:22 206.186.79.9:4349 -> MY.NET.209.231:53 SYN **S*****
Sep 10 00:04:22 206.186.79.9:4350 -> MY.NET.209.232:53 SYN **S*****

Some scans for MOUNTD, which is Sans Top 10 # 6

Sep 11 04:58:48 24.180.134.156:1358 -> MY.NET.208.66:635 SYN **S*****
Sep 11 05:16:40 24.180.134.156:1533 -> MY.NET.208.221:635 SYN **S*****
Sep 11 04:50:37 24.180.134.156:1728 -> MY.NET.208.25:635 SYN **S*****
Sep 11 05:18:10 24.180.134.156:1731 -> MY.NET.208.233:635 SYN **S*****

This is another form of proxy scanning looking for open proxies or proxies that are vulnerable to exploit.
See CERT CA-98.03

Sep 11 19:01:17 168.187.26.157:1068 -> MY.NET.25.203:1080 SYN **S*****

Sep 11 19:01:17 168.187.26.157:1069 -> MY.NET.25.204:1080 SYN **S*****
Sep 11 18:54:07 168.187.26.157:1070 -> MY.NET.17.104:1080 SYN **S*****
Sep 11 19:15:40 168.187.26.157:1070 -> MY.NET.54.169:1080 SYN **S*****

SNMP scans which are Sans Top 10 # 10

Aug 17 11:50:17 134.28.9.225:1745 -> MY.NET.111.67:161 SYN **S*****
Aug 15 17:07:34 195.57.243.171:62286 -> MY.NET.6.7:161 SYN **S*****
Aug 15 17:20:27 195.57.243.171:63782 -> MY.NET.60.8:161 SYN **S*****
Aug 28 15:40:47 198.62.155.106:42769 -> MY.NET.217.10:161 SYN **S*****
Aug 28 15:32:30 207.236.3.96:2938 -> MY.NET.20.10:161 SYN **S*****
Sep 11 05:17:18 24.180.134.156:1196 -> MY.NET.208.226:161 SYN **S*****
Sep 11 05:09:13 24.180.134.156:1479 -> MY.NET.208.154:161 SYN **S*****

Some crafted packet scanning from 24.113.80.28

09/01-08:29:55.509136 24.113.80.28:2439 -> MY.NET.207.34:2272
09/01-08:32:50.481773 24.113.80.28:2439 -> MY.NET.207.34:2272
09/01-08:34:42.373381 24.113.80.28:2439 -> MY.NET.207.34:2272

Sep 6 10:26:49 24.113.80.28:134 -> MY.NET.203.110:1868 UNKNOWN *1***PAU RESERVEDBITS
Sep 6 09:52:26 24.113.80.28:1868 -> MY.NET.203.110:1461 FULLXMAS 21SFRPAU RESERVEDBITS

Section 4

Top Destination IP Scans and Alerts

The repeated access to the same IP and Port could indicate an open proxy server.

08/15-13:50:10.178626 [**] WinGate 1080 Attempt [**] 207.175.201.141:2760 -> MY.NET.60.16:1080
08/15-13:50:11.076192 [**] WinGate 1080 Attempt [**] 207.175.201.141:2760 -> MY.NET.60.16:1080
08/15-13:50:11.374708 [**] WinGate 1080 Attempt [**] 216.67.50.180:1836 -> MY.NET.60.16:1080
08/15-14:28:03.391092 [**] WinGate 1080 Attempt [**] 216.67.50.180:2019 -> MY.NET.60.16:1080
08/15-14:28:03.954172 [**] WinGate 1080 Attempt [**] 207.175.201.141:2927 -> MY.NET.60.16:1080
08/15-14:28:04.071094 [**] WinGate 1080 Attempt [**] 216.67.50.180:2019 -> MY.NET.60.16:1080
08/15-14:28:04.788578 [**] WinGate 1080 Attempt [**] 216.67.50.180:2019 -> MY.NET.60.16:1080

08/15-14:28:05.582451 [**] WinGate 1080 Attempt [**] 207.175.201.141:2927 -> MY.NET.60.16:1080
08/16-21:44:50.444162 [**] WinGate 1080 Attempt [**] 207.175.201.164:1302 -> MY.NET.60.11:1080
08/16-21:44:50.649614 [**] WinGate 1080 Attempt [**] 216.67.82.151:1717 -> MY.NET.60.11:1080
08/16-21:44:50.927425 [**] WinGate 1080 Attempt [**] 166.62.4.48:4105 -> MY.NET.60.11:1080
08/16-21:44:50.966810 [**] WinGate 1080 Attempt [**] 166.90.40.24:1037 -> MY.NET.60.11:1080
08/16-21:44:51.268523 [**] WinGate 1080 Attempt [**] 207.175.201.164:1302 -> MY.NET.60.11:1080
08/16-21:44:51.626155 [**] WinGate 1080 Attempt [**] 166.90.40.24:1037 -> MY.NET.60.11:1080
08/16-21:44:52.228422 [**] WinGate 1080 Attempt [**] 166.90.40.24:1037 -> MY.NET.60.11:1080
08/16-22:36:51.420638 [**] WinGate 1080 Attempt [**] 216.67.82.151:1979 -> MY.NET.60.11:1080
08/16-22:36:51.589952 [**] WinGate 1080 Attempt [**] 166.62.4.48:4329 -> MY.NET.60.11:1080
08/16-22:36:52.655137 [**] WinGate 1080 Attempt [**] 207.175.201.164:1521 -> MY.NET.60.11:1080
08/16-22:36:53.087614 [**] WinGate 1080 Attempt [**] 166.62.4.48:4329 -> MY.NET.60.11:1080
08/18-03:04:47.809448 [**] WinGate 1080 Attempt [**] 208.240.218.220:2980 -> MY.NET.60.11:1080
08/18-03:10:49.190180 [**] WinGate 1080 Attempt [**] 202.188.94.220:2066 -> MY.NET.60.11:1080
08/18-08:22:11.617949 [**] WinGate 1080 Attempt [**] 216.179.0.37:3491 -> MY.NET.60.11:1080
08/18-08:30:22.387463 [**] WinGate 1080 Attempt [**] 209.10.218.250:4582 -> MY.NET.60.11:1080

208.240.218.94 can be found scanning other at http://www.pointman.org/PMFirewall/list-archive/1330.html

Aug 22 22:07:30 ptorico kernel: Packet log: input ACCEPT ppp0
> PROTO=6 208.240.218.220:2332 206.173.24.168:1080 L=60 S=0x00
> I=31240 F=0x4000 T=51 SYN (#38)
> Aug 22 22:07:30 ptorico kernel: Packet log: input DENY ppp0
> PROTO=6 208.240.218.220:2358 206.173.24.168:23 L=60 S=0x00
> I=31291 F=0x4000 T=51 SYN (#14)
> Aug 22 22:07:33 ptorico kernel: Packet log: input DENY ppp0
> PROTO=6 208.240.218.220:2358 206.173.24.168:23 L=60 S=0x00
> I=31544 F=0x4000 T=51 SYN (#14)
> Aug 22 22:07:39 ptorico kernel: Packet log: input DENY ppp0
> PROTO=6 208.240.218.220:2358 206.173.24.168:23 L=60 S=0x00
> I=32213 F=0x4000 T=51 SYN (#14)
> Aug 22 22:07:51 ptorico kernel: Packet log: input DENY ppp0
> PROTO=6 208.240.218.220:2358 206.173.24.168:23 L=60 S=0x00
> I=33467 F=0x4000 T=51 SYN (#14)
> Aug 22 22:08:16 ptorico kernel: Packet log: input DENY pp

MY.NET.60.8 receives a fair bit of alert activity including the following note worthy activity. Among the activity are proxy scans, Watch List activity to the telnet

port and an NMAP finger print scan to attempt and figure out what operating machine the machine is running.

8/18-08:39:40.090940 [**] WinGate 1080 Attempt [**] 207.175.201.144:2354 -> MY.NET.60.8:1080
08/18-08:39:41.747940 [**] WinGate 1080 Attempt [**] 207.175.201.144:2354 -> MY.NET.60.8:1080
08/18-08:39:42.674518 [**] WinGate 1080 Attempt [**] 207.175.201.144:2354 -> MY.NET.60.8:1080
08/18-10:16:32.296254 [**] WinGate 1080 Attempt [**] 216.179.0.37:1096 -> MY.NET.60.8:1080
08/18-16:43:22.859347 [**] WinGate 1080 Attempt [**] 216.67.82.151:2156 -> MY.NET.60.8:1080
08/18-16:43:24.939913 [**] WinGate 1080 Attempt [**] 216.67.82.151:2156 -> MY.NET.60.8:1080
08/11-02:23:44.834384 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1054 -> MY.NET.60.8:23
08/11-02:23:45.619149 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1054 -> MY.NET.60.8:23
08/11-02:23:51.757416 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1054 -> MY.NET.60.8:23
08/16-01:42:07.441198 [**] WinGate 1080 Attempt [**] 207.151.147.201:1632 -> MY.NET.60.8:1080
08/16-01:42:30.220172 [**] Null scan! [**] 207.151.147.201:58190 -> MY.NET.60.8:21
08/16-01:42:30.229811 [**] Probable NMAP fingerprint attempt [**] 207.151.147.201:58191 -> MY.NET.60.8:21
08/16-01:42:30.231975 [**] NMAP TCP ping! [**] 207.151.147.201:58192 -> MY.NET.60.8:21

MY.NET.253.41 appears to be operating as a mail server for an IP on your watch list or at least receives a lot of activity to the e-mail port The traffic from 113 (auth) could be part of a sendmail auth request.

8/11-01:54:40.091429 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:1758 -> MY.NET.253.41:25
08/11-01:54:40.777657 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:1758 -> MY.NET.253.41:25
08/11-01:54:40.777657 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:1758 -> MY.NET.253.41:25
08/11-01:54:41.576859 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:113 -> MY.NET.253.41:55169
08/11-01:54:41.576859 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:113 -> MY.NET.253.41:55169
08/11-01:54:42.276646 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:113 -> MY.NET.253.41:55169

207.114.4.46 performs a large scan for proxy server

08/15-00:44:02.804542 [**] WinGate 1080 Attempt [**] 207.114.4.46:4186 -> MY.NET.60.8:1080
08/15-08:35:27.194847 [**] WinGate 1080 Attempt [**] 207.114.4.46:2175 -> MY.NET.60.11:1080

But not just to you

02/29/2000 13:52:53.592 TCP connection dropped 207.114.4.46, 4138, WAN my.ip, 1080, LAN 'Socks' 0
02/29/2000 13:52:53.608 TCP connection dropped 207.114.4.46, 4139, WAN my.ip, 23, LAN 'Telnet' 0

But since the name resolves to ProxyScan.MD.US.Undernet.Org your problem is not really a rogue proxy scanner but users accessing IRC if this is against your

AUP. They perform this service looking for people connecting through open proxies to anonymize their IRC sessions.

212.179.58.174 is communicating with a watch list IP for what based on a port number of 6699 could be napster. More information on the security implications of napster are available at http://www.sans.org/infosecFAQ/napster.htm

09/14-07:41:27.899852 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.174:2173 -> MY.NET.157.200:6699
09/14-07:41:28.282503 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.174:2173 -> MY.NET.157.200:6699
09/14-07:41:28.282503 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.174:2173 -> MY.NET.157.200:6699
09/14-07:41:28.832247 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.174:2173 -> MY.NET.157.200:6699

Section 5

Some More Watchlist activity

As previously stated, the watch list activity is best analyzed by your own staff that has a better understanding of why the networks in China and Israel have been flagged. Some watchlist activity is already flagged else where in the report.

This could be a napster attempt.

08/17-12:45:38.344565 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.66.2:4807 -> MY.NET.181.87:6699
Repeats Multiple Times

This could be a Trojan scan.

8/20-09:06:02.823194 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.29.150:1098 -> MY.NET.53.28:4407
08/20-09:06:02.823286 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.29.150:1098 -> MY.NET.53.28:4407

08/18-07:25:15.742632 [**] Watchlist 000222 NET-NCFC [**] 159.226.63.190:113 -> MY.NET.253.43:41377

09/02-03:18:55.541479 [**] Watchlist 000222 NET-NCFC [**] 159.226.228.1:113 -> MY.NET.253.42:48917

Repeats Multiple Times Trojan Data Stream?
08/20-15:13:38.125651 [**] Watchlist 000222 NET-NCFC [**] 159.226.114.129:37268 -> MY.NET.162.199:1097
08/20-15:13:38.128313 [**] Watchlist 000222 NET-NCFC [**] 159.226.114.129:37268 -> MY.NET.162.199:1097

Possible Trojan Scanning or telnet session
08/11-16:12:36.219622 [**] Watchlist 000222 NET-NCFC [**] 159.226.41.166:23 -> MY.NET.60.11:10593

08/11-16:12:36.220506 [**] Watchlist 000222 NET-NCFC [**] 159.226.41.166:23 -> MY.NET.60.11:10593
08/11-16:12:36.221111 [**] Watchlist 000222 NET-NCFC [**] 159.226.41.166:23 -> MY.NET.60.11:10593

Possible Peculiar Packet Problems Present Paradox

There are several examples of machine on your network creating out of spec packets, where the normal rules that govern how a packet is formed are not being followed. This can indicate several things, such as attempts to finger print machines based on there response to the crafted packets, broken hard ware, as in the case 195.11.x.x Demon Net in the UK.

On your network many of these packets come from subnets in the MY.NET.2XX.X segments, it is possible that there are some broken network components on these subnetsThese machines should be reviewed to possibly find the cause of the packets.

MY.NET.208.178 send a packet from a source port of 0 to 131.173.27.79 with an SYN/FIN/ACK. This could be a broken network at your site.

09/04-11:22:42.146393 MY.NET.208.178:0 -> 131.173.27.79:2219
TCP TTL:126 TOS:0x0 ID:2361 DF
*1SF**A* Seq: 0x1A2B276A Ack: 0xC47F001A Win: 0x5010
TCP Options => EOL EOL

MY.NET.222.110 connects multiple IP addresses using strange IP combinations

09/04-11:33:02.856184 MY.NET.222.110:6699 -> 172.136.55.69:4504
TCP TTL:126 TOS:0x0 ID:33099 DF
SFR* Seq: 0x2EA860A Ack: 0x2E4 Win: 0x5010
1A 2B 11 98 02 EA 86 0A 00 00 02 E4 08 07 50 10 .+..........P.
21 2C 5A 43 00 00 CA E0 38 18 70 78 E8 88 24 36 !,ZC....8.px..$6
F7 4C

SECTION 6

Miscellaneous

While these first logs indicate possible use of ICQ since the IP address is registered to icq.aol the second logs indicate possible use of NAPSTER where the port is 6699, .In many cases the real issue with type of activity are possible violations of your organizations Acceptable Use Policy.

ICQ

09/11-20:35:37.773145 [**] Attempted Sun RPC high port access [**] 205.188.153.115:4000 -> MY.NET.218.218:32771
09/11-20:35:37.773145 [**] Attempted Sun RPC high port access [**] 205.188.153.115:4000 -> MY.NET.218.218:32771
09/11-20:35:37.773145 [**] Attempted Sun RPC high port access [**] 205.188.153.115:4000 -> MY.NET.218.218:32771
09/11-20:35:37.773145 [**] Attempted Sun RPC high port access [**] 205.188.153.115:4000 -> MY.NET.218.218:32771

NAPSTER
8/28-10:00:32.123451 MY.NET.202.202:6699 -> 152.2.167.91:1731
08/28-10:01:23.356403 MY.NET.202.202:6699 -> 152.2.167.91:1735
08/28-16:09:36.670407 129.93.204.53:6699 -> MY.NET.217.194:1135
08/29-10:11:54.694547 24.114.90.7:6699 -> MY.NET.204.86:1518
08/29-15:43:15.635554 193.150.240.95:6699 -> MY.NET.110.192:1134
08/29-16:40:58.689489 24.29.7.227:6699 -> MY.NET.204.190:2167
08/29-16:44:46.387591 24.29.7.227:6699 -> MY.NET.204.190:2169
08/29-21:09:31.176498 128.175.127.173:6699 -> MY.NET.217.30:3316
08/29-21:26:59.974931 MY.NET.219.30:6699 -> 129.81.91.208:1316

Trojan Scanning

Trojan scanning is rampant on the Internet. Included is evidence of people scanning for the most common Trojans, however, a scan does not a node penetration make. Most of these scans are untargeted and looking for already compromised machines. These represent only a select few of the many probes for these Trojans and they are marked since they correspond to well known ports for Trojans.

Hack A Tack
Sep 9 17:34:21 147.208.171.139:2737 -> MY.NET.97.230:31789 SYN **S*****
SUB7
Sep 9 17:33:30 147.208.171.139:2202 -> MY.NET.97.230:27374 SYN **S*****
Back Orifice
Aug 17 16:37:09 147.208.171.139:3498 -> MY.NET.150.89:31337 SYN **S*****

SECTION 7

Recommendations

Without meeting with your staff to go over the results it is difficult to make sound recommendations, however, here are some point for you to consider.

- Your watchlist generates an very large amount of traffic, if you have no need for communicating with these networks they could be blocked at your router
- Your network is very active, an additional IDS behind any firewall protected LANs would help in weeding out the serious threats that make it across your firewall.
- Host based firewalls or host based IDS tools can help you determine if an attempt an attack was successful.
- A penetration test of the network for known common vulnerabilities would help you interpret these IDS logs against known soft spots in your architecture.

Last Updated: July 28th, 2017

Upcoming Training

SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced